Group Member: Mingjun Wen, Boyu Yu
PA1 Part2 Report

Final Schema:

```
CREATE TABLE Users(
   user_id int4 AUTO_INCREMENT,
   first_name VARCHAR(20),
   last_name VARCHAR(20),
   email varchar(50) UNIQUE,
   birth_date DATE,
   hometown VARCHAR(20),
   gender VARCHAR(10),
   password varchar(30) NOT NULL,
        PRIMARY KEY (user_id)
);

CREATE TABLE Albums(
        album_id int4 AUTO_INCREMENT,
        album_name VARCHAR(50) NOT NULL,
        date DATE,
        user_id INTEGER NOT NULL,
        PRIMARY KEY (album_id),
        FOREIGN KEY (user_id) REFERENCES Users(user_id)
);

CREATE TABLE Pictures(
        picture_id int4 AUTO_INCREMENT,
   album_id INTEGER NOT NULL,
        user_id INTEGER NOT NULL,
        num_like INTEGER DEFAULT 0,
        imgdata longblob,
        caption VARCHAR(255),
        PRIMARY KEY (picture_id),
        FOREIGN KEY (user_id) REFERENCES Users(user_id),
   FOREIGN KEY (album_id) REFERENCES Albums(album_id)
);
```

```
CREATE TABLE Comments(
        comment_id int4 AUTO_INCREMENT,
        picture_id INTEGER NOT NULL,
        user_id INTEGER NOT NULL,
        comment_text VARCHAR(100),
        comment_date DATE,
        PRIMARY KEY (comment_id),
        FOREIGN KEY (user_id) REFERENCES Users(user_id),
        FOREIGN KEY (picture_id) REFERENCES Pictures(picture_id)
);

CREATE TABLE Likes(
        picture_id INTEGER NOT NULL,
        user_id INTEGER NOT NULL,
        PRIMARY KEY (picture_id,user_id),
        FOREIGN KEY (user_id) REFERENCES Users(user_id),
        FOREIGN KEY (picture_id) REFERENCES Pictures(picture_id)
);

CREATE TABLE Friends(
        user_id1 INTEGER NOT NULL,
        user_id2 INTEGER NOT NULL,
        PRIMARY KEY (user_id1, user_id2),
        FOREIGN KEY (user_id1) REFERENCES Users(user_id),
        FOREIGN KEY (user_id2) REFERENCES Users(user_id)
);

CREATE TABLE FriendRequest(
        sender_id INTEGER NOT NULL,
        recipient_id INTEGER NOT NULL,
        message VARCHAR(150),
        PRIMARY KEY(sender_id, recipient_id),
        FOREIGN KEY (sender_id) REFERENCES Users(user_id),
        FOREIGN KEY (recipient_id) REFERENCES Users(user_id)
);

CREATE TABLE Tags(
        tag_id int4 AUTO_INCREMENT,
        name VARCHAR(50) UNIQUE,
        PRIMARY KEY (tag_id)
);
```

```
CREATE TABLE Tagged(
        tag_id INTEGER,
        picture_id INTEGER,
        PRIMARY KEY (tag_id, picture_id),
        FOREIGN KEY (tag_id) REFERENCES Tags(tag_id),
        FOREIGN KEY (picture_id) REFERENCES Pictures(picture_id)
);
```

Additional Assumptions and constraints:

We assumed that every user has to create an album before they submit any photo, because photos need to belong to one of the albums, otherwise the user will not be able to upload the photo into our website. Without creating the album, the photo will not be able to upload. This constraint is reinforced in the app.py.

Upon uploading photos, users can choose to add any amount of tags to their photo. When inputting their photos, they are required to separate each tag with a comma (,).

The user cannot send a friend request to him/herself. Doing so the application will interrupt by displaying a warning. Two users become friends once both of them send a friend request to each other. Users cannot send friend requests to their friends. Doing so will trigger a warning. This constraint is reinforced by the app.py.

In order to like a photo, the user has to enter the photo ID of the picture instead of having all the images displayed and select which one to like. This is also applied to comments on a photo, where the user has to input the photo ID of the picture that he/she wants to comment on.

When there are not enough users or there aren't any Friends of Friends available, the *Friends of Friends Recommendation* won't display anything. Please make sure that there exists at least one user who is the friend of the friend of the testing account when testing the program. This constraint is reinforced by app.py.

In order to use *Search Photo by Tags* function, the user has to have at least an tag created and associated to at least one photo. Otherwise the application won't let the user search using tags. When searching a photo using tags, the user can only input one tag. This constraint is reinforced by the app.py.